



WTMI 5

CONTATO

cauainocencio14@hotmail.com¹

Wisleycesar032@gmail.com²



INSTITUTO FEDERAL
Goiano
Campus Morrinhos

ALGORITMOS DE ORDENAÇÃO EM JOGOS: A MELHORIA DA JOGABILIDADE E EXPERIÊNCIA DO JOGADOR

Cauã Cristian Inocência¹, Wisley César Borges do Vale²

INTRODUÇÃO

Os videogames evoluíram para plataformas complexas que lidam com grandes volumes de dados, desde informações de jogadores até placares e objetivos específicos. Uma das técnicas fundamentais que permitem que os jogos gerenciem esses dados de maneira eficiente é a ordenação. Esta pesquisa examina a utilização dos algoritmos de ordenação em videogames, concentrando-se em como esses algoritmos são empregados para classificar jogadores, placares e metas no jogo.

```
PS C:\Users\caua\Desktop\sla> dart "C:\Users\caua\Desktop\sla\quicksort.dart"
Insira o tamanho da lista:
100
Tempo para ordenar: 0 ms
PS C:\Users\caua\Desktop\sla> dart "C:\Users\caua\Desktop\sla\quicksort.dart"
Insira o tamanho da lista:
1000
Tempo para ordenar: 1 ms
PS C:\Users\caua\Desktop\sla> dart "C:\Users\caua\Desktop\sla\quicksort.dart"
Insira o tamanho da lista:
10000
Tempo para ordenar: 2 ms
PS C:\Users\caua\Desktop\sla> dart "C:\Users\caua\Desktop\sla\quicksort.dart"
Insira o tamanho da lista:
100000
Tempo para ordenar: 13 ms
PS C:\Users\caua\Desktop\sla> dart "C:\Users\caua\Desktop\sla\quicksort.dart"
Insira o tamanho da lista:
1000000
Tempo para ordenar: 185 ms
PS C:\Users\caua\Desktop\sla> dart "C:\Users\caua\Desktop\sla\quicksort.dart"
Insira o tamanho da lista:
10000000
Tempo para ordenar: 7327 ms
```

Figura 2. Tempo de execução do QuickSort.

OBJETIVO

O propósito deste estudo é identificar e analisar o principal algoritmo de ordenação utilizado em videogames, buscando compreender sua relevância na otimização da experiência do jogador e no funcionamento interno do jogo.

METODOLOGIA

Foram escolhidas plataformas e sites reconhecidos no campo da ciência da computação, incluindo GitHub, Stack Overflow e fóruns especializados, para identificar implementações práticas do algoritmo de ordenação QuickSort e BubbleSort. Os testes ocorreram em uma máquina padrão e os algoritmos foram implementados em Dart. Uma quantidade padrão de dados foi usado para avaliação, e o menor tempo de execução foi registrado para comparação.

```
Insira o tamanho da lista:
100
Tempo para ordenar: 1 ms
PS C:\Users\caua\Desktop\sla> dart "C:\Users\caua\Desktop\sla\bubblesort.dart"
Insira o tamanho da lista:
1000
Tempo para ordenar: 1 ms
PS C:\Users\caua\Desktop\sla> dart "C:\Users\caua\Desktop\sla\bubblesort.dart"
Insira o tamanho da lista:
10000
Tempo para ordenar: 123 ms
PS C:\Users\caua\Desktop\sla> dart "C:\Users\caua\Desktop\sla\bubblesort.dart"
Insira o tamanho da lista:
100000
Tempo para ordenar: 15417 ms
```

Figura 1. Tempo de execução do BubbleSort.

CONCLUSÃO

A partir da análise e dos testes realizados, foi possível concluir que o algoritmo de ordenação QuickSort demonstrou mais eficiência em termos de tempo de execução em comparação ao BubbleSort em grandes cenários. Em cenários de grande volume de dados, como é encontrado em videogames é essencial o uso de um algoritmo de ordenação que suporta essas grandes quantidades com eficiência. Ao comparar com um algoritmo básico como BubbleSort, observou-se que o tempo de execução aumentou exponencialmente com o tamanho do conjunto de dados, tornando-se impraticável para aplicações em tempo real, como jogos. Essas observações reforçam a importância da escolha adequada dos algoritmos no desenvolvimento de videogames, garantindo tanto a eficiência técnica quanto a otimização da experiência do usuário.

REFERÊNCIAS

- Dutta, U. (Jul 16, 2020). Dart sorting algorithms: Quick sort. Medium. <https://medium.com/@utsavdutta/dart-sorting-algorithms-quick-sort-e9a9ad9c1496>.
- Dutta, U. (Jul 9, 2020). Dart sorting algorithms: Bubble sort. Medium. <https://medium.com/@utsavdutta/dart-sorting-algorithms-bubble-sort-9170cf2e77ca>